The GAN that Warped: Semantic Attribute Editing with Unpaired Data

Garoe Dorta [1,2] Sara Vicente [2] Neill D. F. Campbell [1] Ivor Simpson [2]
[1]University of Bath [2]Anthropics Technology Ltd.
[1]{g.dorta.perez,n.campbell}@bath.ac.uk [2]{sara,ivor}@anthropics.com

(a) Input image (previously unseen) (b) User requested edit: "big nose" (c) User requested edit: "narrowed eyes"

Figure 1: An illustrative example of semantic image editing at high resolution (3456×5184). The user only requests a change in a semantic binary attribute and the input image (a) is automatically transformed into, *e.g.*, an image with a "big nose" (b) or "narrowed eyes" (c). The identity and high resolution detail of the original input is preserved. Image courtesy of flickr user Kaue Lima.

Abstract

Deep neural networks have recently been used to edit images with great success. However, they are often limited by only being able to work at a restricted range of resolutions. They are also so flexible that semantic face edits can often result in an unwanted loss of identity. This work proposes a model that learns how to perform semantic image edits through the application of smooth warp fields. This warp field can be efficiently predicted at a reasonably low resolution and then resampled and applied at arbitrary resolutions. Previous approaches that attempted to use warping for semantic edits required paired data, that is example images of the same object with different semantic characteristics. In contrast, we employ recent advances in Generative Adversarial Networks that allow our model to be effectively trained with unpaired data. We demonstrate the efficacy of our method for editing face images at very high resolutions (4k images) with an efficient single forward pass of a deep network at a lower resolution. We illustrate how the extent of our edits can be trivially reduced or exaggerated by scaling the predicted warp field, and we also show that our edits are substantially better at maintaining the subject's identity.

1. Introduction

Face editing has a long history in computer vision [23, 26, 34] and has been made increasingly relevant with the rise in the number of pictures people take of others or themselves. The type of edits that are desired are usually of semantic nature, such as removing a moustache or changing

the subject's expression from a frown to a smile.

In the last few years, deep learning approaches have become the standard in most editing tasks, including inpainting [27], super-resolution [21], and face editing [29]. Recently image-to-image translation methods such as [16] have been proposed, which learn how to transform an image from a source domain to a target domain. The recently proposed Cycle-GAN approach [40] allows learning such translations from unpaired data, *i.e*. for each source image in the dataset a corresponding target image is not required.

In this paper we are interested in photo-realistic image editing, which is a subset of image-to-image translation. We also focus on methods that provide a simple interface for users to edit images, *i.e*. a single control per semantic characteristic [6, 30], as this makes it easier for novice users.

Most deep learning methods predict the pixel values of an edited image directly [6, 7, 29, 30]. A consequence of this approach is that these methods are limited to only being effective on images that have a similar resolution to the training data. A further disadvantage of current methods is the difficulty of applying a partial or exaggerated edit as opposed to a binary attribute change. For this to be possible, an extensive collection of soft attribute data is required, which is labor intensive, and at test time each intermediate value requires another forward pass of the network, creating increased computational expense. [30].

Recently, some interesting approaches that do allow edits at higher resolutions have been proposed. They proceed by estimating the edits at a fixed resolution and then applying them to images at a higher resolution. The types of possible edits are restricted to either warping [37] or local linear color transforms [10]. However, these approaches are limited by requiring paired data, *i.e*. for each source image in the dataset, they need the corresponding edited image.

Inspired by these high resolution methods, in this paper we introduce an approach to learn warp fields for semantic image editing without the requirement of paired training data samples. This is achieved by exploiting recent approaches for learning edits from unpaired data with cycle-consistency checks. Our proposed model uses a similar framework to StarGAN [6] to predict warp fields that apply the requested edits. As the predicted warp fields are smooth, they can be trivially upsampled and applied at high resolutions.

A potential criticism is that there are clear limitations to the types of edits possible through warping. We argue that, for the changes that *can* be described in this way, there are several distinct benefits. The advantages of our proposed model with respect to pixel translation models can be summarized as:

1. Smooth warp fields can be trivially upsampled and applied to higher resolution images with a minimal loss of fidelity. This is opposed to upsampling photographs, which commonly contain high frequency . We demonstrate this benefit by applying the warps at a higher resolution than they were estimated at, e.g. fig. 1.
2. Warp field models are a constrained type of pixel translation models. Such constrained models are easier to learn and priors can be added to regularise against unrealistic edits. We demonstrate that restricting edits to smooth warp fields leads to a model that is better at preserving a subject's identity.
3. Warp fields are more interpretable than pixelwise differences, particularly with respect to identifying potentially erroneous or unrealistic edits. We illustrate this by providing maps showing regions where the image has been overly stretched or squashed, resulting in unrealistic local textures.
4. Warp fields are much more suited to allow partial edits than pixel based approaches. We demonstrate the simplest implementation of this by scaling the warp field to show interpolation and extrapolation, and qualitatively show results that are plausible.
5. Editing most of the image via warping, allows us to inpaint models at much higher resolution in areas of limited size, where a divergent warp would reveal previously unseen content.

We demonstrate the efficacy of our method by providing quantitative and qualitative results in the domain of faces by manipulating expressions and other semantic attributes.

2. Previous work

This work builds upon recent work in image-to-image translation. These models learn to modify the semantic characteristics of an image. Our novelty is in describing these edits as smooth deformation fields, rather than producing an entirely new image. Such deformation fields can be upsampled and therefore the edits can be applied at arbitrary resolutions. Some previous works that allow high resolution editing reply upon paired data examples or require costly optimisation, rather than a single forward pass of a network; neither of which is required for the proposed approach. An overview of the characteristics of our work compared to previous methods is shown in Table 1.

2.1. Image-to-Image translation

Image-to-Image translation models, such as Pix2Pix [16], learn to transform an image from a source domain to a target domain using an adversarial loss [12]. This approach requires paired training data; *i.e*. each image in the source domain must have a corresponding image in the target domain. Given this restriction, the method is often applied to problems where collecting paired data is easier, such as colorization, or semantic labels to RGB.

Method	Unpaired Data	High Resolution	Forward Pass
Pix2Pix [16]			✓
CycleGAN [40]	✓		✓
StarGAN [6]	✓		✓
FaceShop [29]	✓		✓
FlowVAE [37]		✓	✓
CWF [9]		∼	✓
DBL [10]		✓	✓
iGAN [39]	✓	∼	
DFI [35]	✓	∼	
Ours	✓	✓	✓

Table 1: Compared to previous work on image-to-image translation, our method is the only one that is able to edit high-resolution images in a forward pass of the network, without paired training data. The symbol ∼ denotes partial fulfilment of a criterion.

Several extensions of Pix2Pix have been proposed that perform image-to-image translation without requiring paired data. In Cycle-GANs [40], two generators are trained, one from source to target domain and vice versa, with a cycle-consistency loss on the generation process. However, this does not scale well with an increase in the number of domains. StarGAN [6] addresses this issue by conditioning the generator on a domain vector, and adding a domain classification output layer to the discriminator.

2.2. Editing of high resolution images

Methods for editing images at high resolution can be divided in two categories: (i) methods that use intermediate representations that are designed to upsample well to arbitrary resolutions, and (ii) methods that directly predict pixel values at high resolutions.

Methods designed for upsampling These approaches are based on predicting intermediate representations that are relatively agnostic to image resolution; *e.g*. warp fields, local color affine transformations or blendshape weights.

Warp fields, if sufficiently smooth, can be predicted at a lower resolution, upsampled and applied at high resolution with minimal loss of accuracy. Previous methods have applied them to redirecting eye gaze [9] and editing emotional expressions [37]. However, both of these methods require paired training data.

Local affine color transformations [5, 10] has also been predicted based on paired low resolution images and effectively applied to the original resolution. Although these methods have limited capacity for making semantic changes, and are more suited for image enhancement.

While previous methods directly predict the intermediate representations, Zhu *et al*. [39] train a low-resolution GAN and then fit a dense warp field and local affine color transformation to a pair of input-output images. The network is unaware that these restricted transformations will be fitted to its outputs, so capacity is potentially wasted learning edits that are not representable by such transformations.

Blendshape weights have also been used as an intermediate representation to edit expressions in the context of video reenactment [33, 24]. Similar to our approach, the blendshape weights are resolution independent. However, these methods require several input video frames to reconstruct the face for the blendshape model.

Direct prediction at high resolution A number of techniques have been proposed in order to scale deep image synthesis methods to larger image resolutions. These include, synthesizing images in a pyramid of increasing resolutions [8], employing fully convolutional networks trained on patches [21], and directly in full resolution [3, 17]. However, direct or pyramid based approaches do not scale well beyond modest resolutions and training on patches assumes that global image information is not needed. Image editing applications using the aforementioned methods have also been explored [15, 29].

A gradient descent based method for image editing was proposed in [35]. The image is modified by following gradient directions of a pretrained classification network, until it is classified as having the desired attributes. This approach fails when the input resolution differs significantly from the input data. Furthermore, the fact that gradient descent is performed at test time limits its applicability (generating an 1000×1000 image takes approximately 2 minutes).

3. Background

Generative Adversarial Network (GAN) [12] models consist of two parts, a generator and a discriminator. The discriminator classifies data as real or fake, where it is trained with the real examples drawn from a training set and the fake examples as the output of the generator. The generator is trained to fool the discriminator into classifying generated samples as real. Formally, a GAN is defined by the following min-max game objective:

$$\min_G \max_D \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})} [\log(D(\mathbf{x}))] + \mathop{\mathbb{E}}_{\mathbf{z}\sim p(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))], \quad (1)$$

where $\mathbf{x}$ denotes input data from an empirical distribution $p_{\text{data}}(\mathbf{x})$ (the dataset), $\mathbf{z}$ is a random variable drawn from an arbitrary distribution $p(\mathbf{z})$, G is the generator and D is the discriminator.

Given two data domains, A and B, Cycle-GAN [40] learns a pair of transformations $G : A \rightarrow B$ and $H : B \rightarrow A$. Unlike previous approaches [16] this does not require paired sample from A and B, but instead utilises a cycle consistency loss ($||\mathbf{x}_a - H(G(\mathbf{x}_a))||_1$, where $\mathbf{x}_a$ is a sample image from domain A) to learn coherent transformations that preserve a reasonable amount of image content. Cycle-

GAN models are limited in that they require 2 generators and 2 discriminators for each domain pair.

Cycle-GAN was generalised by StarGAN [6] to require only a single generator and discriminator to edit multiple domains. Here, each image $\mathbf{x}$ has a set of associated domains, represented as a binary vector $\mathbf{c}$. The generator model, $G(\mathbf{x}, \bar{\mathbf{c}})$, transforms $\mathbf{x}$ to match the target domains indicated by $\bar{\mathbf{c}} \sim p(\mathbf{c})$, where $p(\mathbf{c})$ is the empirical labels distribution. Alongside the traditional GAN adversarial term

$$L_{adv} = \mathbb{E}_{\mathbf{x}} \left[\log(D(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{x},\bar{\mathbf{c}}} \left[\log(1 - D(G(\mathbf{x}, \bar{\mathbf{c}})))\right], \quad (2)$$

StarGAN also uses a cycle consistency loss:

$$L_c = \|\mathbf{x} - G(G(\mathbf{x}, \bar{\mathbf{c}}), \mathbf{c})\|_1 \quad (3)$$

and domain classification losses:

$$L^d_{cls} = -\log(C(\mathbf{x}, \mathbf{c})) \quad (4)$$

$$L^g_{cls} = -\log(C(G(\mathbf{x}, \bar{\mathbf{c}}), \bar{\mathbf{c}})), \quad (5)$$

where $C(\mathbf{x}, \mathbf{c})$ is a discriminative function that outputs the probability that $\mathbf{x}$ has associated domains $\mathbf{c}$. These losses train the classifier using the training set (eq. 4) and ensure the translated image matches the target domains (eq. 5).

4. Methodology

Our goal is to learn image-to-image transformations that can be applied at arbitrary scales without paired training data. An overview of our system is shown in Figure 2.

We employ the StarGAN framework as the basis for our model and use the notation introduced above. We modify the generator such that the set of transformations is restricted to non-linear warps of the input image:

$$G(\mathbf{x}, \bar{\mathbf{c}}) = T(\mathbf{x}, W(\mathbf{x}, \bar{\mathbf{c}})), \quad (6)$$

where T is a predefined warping function that takes as input an image and a non-linear warp and applies the warp to the image, and $W(\mathbf{x}, \bar{\mathbf{c}}) = \mathbf{w}$ is a parametric function that generates the non-linear warps. The family of parametric functions for W is chosen to be a neural network.

4.1. Warp Parametrizations

There are a number of available parameterizations for the non-linear warp fields, $\mathbf{w}$. Two possible approaches are landmark based and dense warps.

Landmark based methods involve defining displacements on several sparsely defined landmarks on the object to be deformed, where a smooth dense warp field can be constructed through the use of an interpolation techniques, such as thin plate splines [2]. Such a parametrisation has the advantage of having a reduced parameter set to predict, making the model easier to train. However, this comes at the cost of reduced deformation flexibility and relies on accurate and robust landmark finding. In preliminary experiments we found this approach too restrictive for our purposes, please refer to Appendix F for results on this model.

Dense warps, in the form of a displacement vector at each pixel, allow for arbitrary deformations. This gives flexibility at the cost of model complexity. Such an approach also gives no guarantees on warp field smoothness, which is crucial for application at arbitrary resolutions. To ensure smoothness regularization terms must be employed. Given the additional flexibility, we choose to use dense warps.

4.2. Landmark Locations

Providing the network with structural landmark locations for the object to be edited provides important shape and pose information to the network. Landmarks can be fed to network by transforming them into heatmap images and concatenating with the input image. Heatmaps can be created by setting all the pixels in a given radius around the landmark location to one. This is followed by blurring to reflect uncertainty in the location.

4.3. Learning

We use the same adversarial loss (eq. 2) and domain classification losses (eq. 5 and eq. 4) as StarGAN. The Wasserstein gradient penalty term of Gulrajani *et al*. [13] is added for stability and denoted as L_{gp}.

Warp specific losses The cycle consistency loss is modified to produce warp fields that are inverse consistent, i.e. the composition of the forward and backward transformations yields an identity transformation:

$$L_c = \|T(T(\mathbf{A}, \mathbf{w}), \bar{\mathbf{w}})) - \mathbf{A}\|_2^2 \quad (7)$$

where $\bar{\mathbf{w}} = W(T(\mathbf{x}, \bar{\mathbf{c}}), \mathbf{c})$, and $\mathbf{A}$ is a two channel image where each pixel takes the value of its coordinates. This loss provides additional guidance to the network in terms of dual learning [32]. Also, this should encourage smoother warps as these are easier to invert .

As we do not have paired samples, the generator network could easily find undesired correlations in the data. Previous work [25, 30] employed attention mechanisms to restrict the extend of the edits. As our model is constrained to edit the image by means of geometric deformations, we found it is sufficient to add a sparsity penalty:

$$L_r = \textstyle\sum_{(i,j)} \|\mathbf{w}_{i,j}\|_1, \quad (8)$$

where $\mathbf{w}_{i,j}$ is the displacement vector at pixel (i, j).

The generator network estimates an independent deformation per pixel. By default, there are no guarantees that the learned warps will be smooth. Therefore, an $L2$ penalty

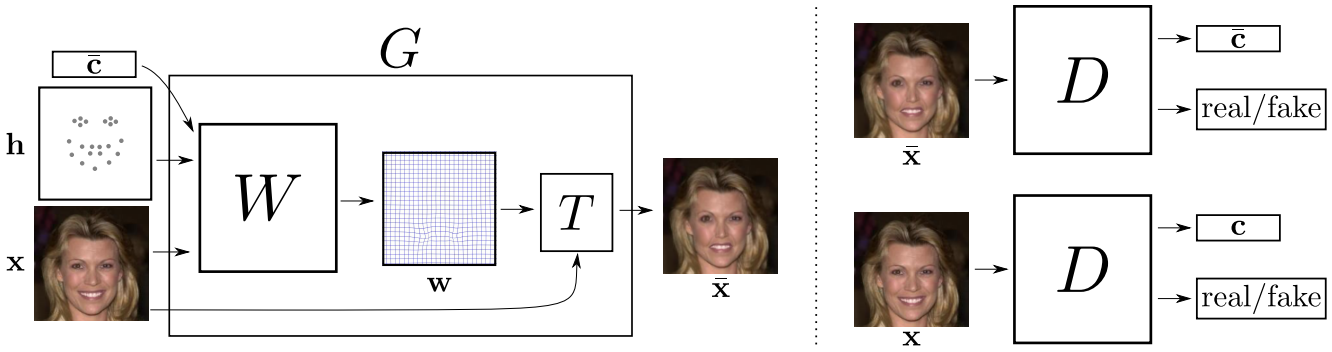

Figure 2: Overview of our warping model, which consists of a generator, G, and a discriminator, D. The inputs to the warping network, W, are an RGB image, $\mathbf{x}$, a set of landmark locations, $\mathbf{h}$, and a one-hot encoded attribute vector, $\bar{\mathbf{c}}$. The output is a dense warp field, $\mathbf{w}$, which can be used by a warping operator, T to deform the input image and produce the output image $\bar{\mathbf{x}}$. The discriminator evaluates for both images, $\mathbf{x}$ and $\bar{\mathbf{x}}$, whether they are real or generated data. It also discriminates whether they contain the labels of their corresponding domain. In this example the source domain contains smiling faces, while the target domain does not.

on the warp gradients is added to encourage smoothness. In practice a finite-difference approximation is used as

$$L_s = \frac{1}{n} \sum_{(i,j)} \|\mathbf{w}_{i+1,j} - \mathbf{w}_{i,j}\|_2^2 + \|\mathbf{w}_{i,j+1} - \mathbf{w}_{i,j}\|_2^2, \quad (9)$$

where n is the number of pixels in the warp field.

The joint losses for the discriminator and the generator are defined as

$$L_D = -L_{adv} + \lambda_{gp} L_{gp} + \lambda_{cls} L_{cls}^d, \quad (10)$$

$$L_G = L_{adv} + \lambda_{cls} L_{cls}^g + \lambda_c L_c + \lambda_r L_r + \lambda_s L_s \quad (11)$$

where λ_{cls}, λ_{gp}, λ_c, λ_r and λ_s are hyper-parameters that control the relative strength of each loss.

4.4. Inference

Once the model parameters have been optimized, an input image of arbitrary size can be edited in a single forward pass of the network.

Using the landmarks of the input image, a global affine transformation is used to align and resize the image to the mean of the training data. The aligned image is fed to the discriminator to estimate its labels. The values for some labels are edited to generate the target labels, such as changing the smile attribute. The low-resolution image and target labels are fed into the generator, which produces a suitable low-resolution warp field, $\mathbf{w}$. The warp field displacement vectors are transformed and bilinearly resampled to the original image resolution, using the inverse of the affine transformation. The original image is warped using the high-resolution warp field to produce the final edited image.

4.5. Face specific considerations

As we demonstrate our model on face data, we present additional considerations when working with these images.

Landmarks Prediction of face landmark locations is a well studied field, and there are a number of off-the-shelf methods [18] to extract points from a face. A set of landmark points are converted into heatmaps, by setting the pixel value at the landmark location to one, blurring it and dividing by the maximum intensity. These heatmaps are concatenated with the image $\mathbf{x}$, and given as input to the generator network. We found this additional supervision in terms of landmarks to improve the quality of the results.

Losses We found, empirically, that the warps might introduce undesirable distortions over the eyes. To prevent these, we impose a stronger smoothness penalty on the eyes

$$L_e = \frac{1}{p} \sum_{(i,j) \in P} \|\mathbf{w}_{i+1,j} - \mathbf{w}_{i,j}\|_2^2 + \|\mathbf{w}_{i,j+1} - \mathbf{w}_{i,j}\|_2^2, \quad (12)$$

where P is the set of pixels in the eyes, and p is the number of pixels in that region. This loss is added to the generator with a weight parameter λ_e.

Mouth region masking Given the adversarial loss, our approach is penalized for producing unrealistic warps. In order to allow our model to stretch the mouth area if needed, we automatically mask out the inner mouth area for all input images. This allows the generator to warp the mouth area, without being penalized by the discriminator. Masks for the mouth and eyes areas are created using the contours of the detected landmarks.

5. Results

Baselines Our main baseline is StarGAN [6] as it is the most similar method to our approach. We also evaluate a variant of iGAN [39], where we fit a dense flow field using the approach in [38], to the results generated by StarGAN, we denote these method by "SG + Flow".

We experimented with the GANimation [30] approach using the code provided by the authors. However we were

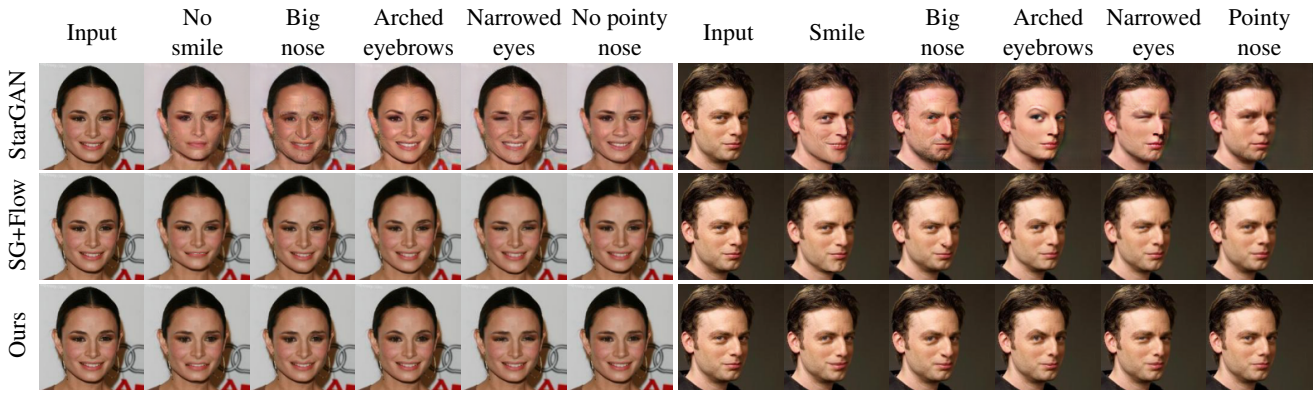

Figure 3: Comparison to previous work methods on the CelebA dataset. From a given input image, first column, each method attempts to transfer the semantic attribute in its corresponding column. Our approach is able to edit the attributes of the input images while better preserving the identity of the subject.

unable to generate meaningful results when training the method with binary attributes. We suspect that this is due to the method's reliance on action unit labels.

Hyper-parameters All models were trained on a single Titan X GPU using the Tensorflow [1] framework. All methods use the Adam optimizer [19] with a learning rate of 0.0001, with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We use the same hyper-parameters for both datasets: $\lambda_{\text{cls}} = 25$, $\lambda_{\text{gp}} = 10$, $\lambda_{\text{c}} = 10$, $\lambda_{\text{r}} = 15$, $\lambda_{\text{s}} = 125$ and $\lambda_{\text{e}} = 500$. In addition, the computed displacements are constrained to a maximum of 5 pixels, by applying a sigmoid non-linearity to the last layer of the generator and scaling the output. Additional implementation details, as well as the model architectures are provided in Appendix B and H.

5.1. Datasets

We evaluate our method and baselines on two face datasets, CelebA [22] and RafD [20].

CelebA The CelebA [22] dataset contains 202,599 images of faces and we use the train/test split recommended by the authors. The faces are center-cropped and resized to 128×128. We employ an internal face landmark detection network to extract 49 points per face. Importantly, from the 40 binary attributes provided, we choose the ones more amenable to be characterized by warping, namely: smiling, big nose, arched eyebrows, narrow eyes and pointy nose.

RafD The RafD [20] dataset contains images of 67 subjects in 8 expressions. For each expression, the subjects were recorded from 5 camera angles and from 3 different eye gaze directions. We discarded the two most extreme camera angles, leaving a total of 4,824 images. Contrary to previous work [6, 30], which kept images from the same subject on the train and test set, we reserve all images of subjects 58, 63, 64, 71 and 72 as test data. Face landmarks were automatically detected in all images, the images aligned to the mean CelebA [22] face and resized to 128×128. We only consider transformations from neutral to all other emotions and vice-versa.

5.2. Qualitative results

We show qualitative results on the CelebA dataset in Fig. 3. For each input image, we show the edited images corresponding to changing individual attributes. StarGAN [6] often edits characteristics of the image that are not related with the changed attribute, such as the skin tone or the background color. StarGAN + Flow generate images that better keep the content, however warping based on optical flow can lead to artefacts when the method fails to find good correspondences. Although, the attribute changes produced by our method are more subtle, they better preserve the identity of the subject.

Qualitative results are shown for the RafD dataset on Fig. 4. As this dataset contains paired data, we can compare the input images with ground truth targets. Moreover, it also allows to employ a standard optical flow technique [38] to directly compute the warps between the same person expressions, which we denote as "Flow" in the figure. For the edits where the expression leads to an open mouth, we employ a simple inpainting model for the mouth area. Details of this inpainting model are given in Appendix I.

For some of the emotions, our model produces more realistic edits than previous work. In addition, we can apply our results seamlessly at the original image resolution of 580×540, denoted as "Ours HR", in contrast to the 128×128 resolution of StarGAN. Fig. 1 demonstrates the power of the warping representation by operating at a far higher resolution (3456×5184) than can be achieved by direct methods.

Another advantage of our model is that once a warp field has been computed for a given input image, we can apply partial edits by simply scaling the predicted displacement vectors by a scalar, α. Results of interpolation and extrapolation of warp fields generated by our model are shown in

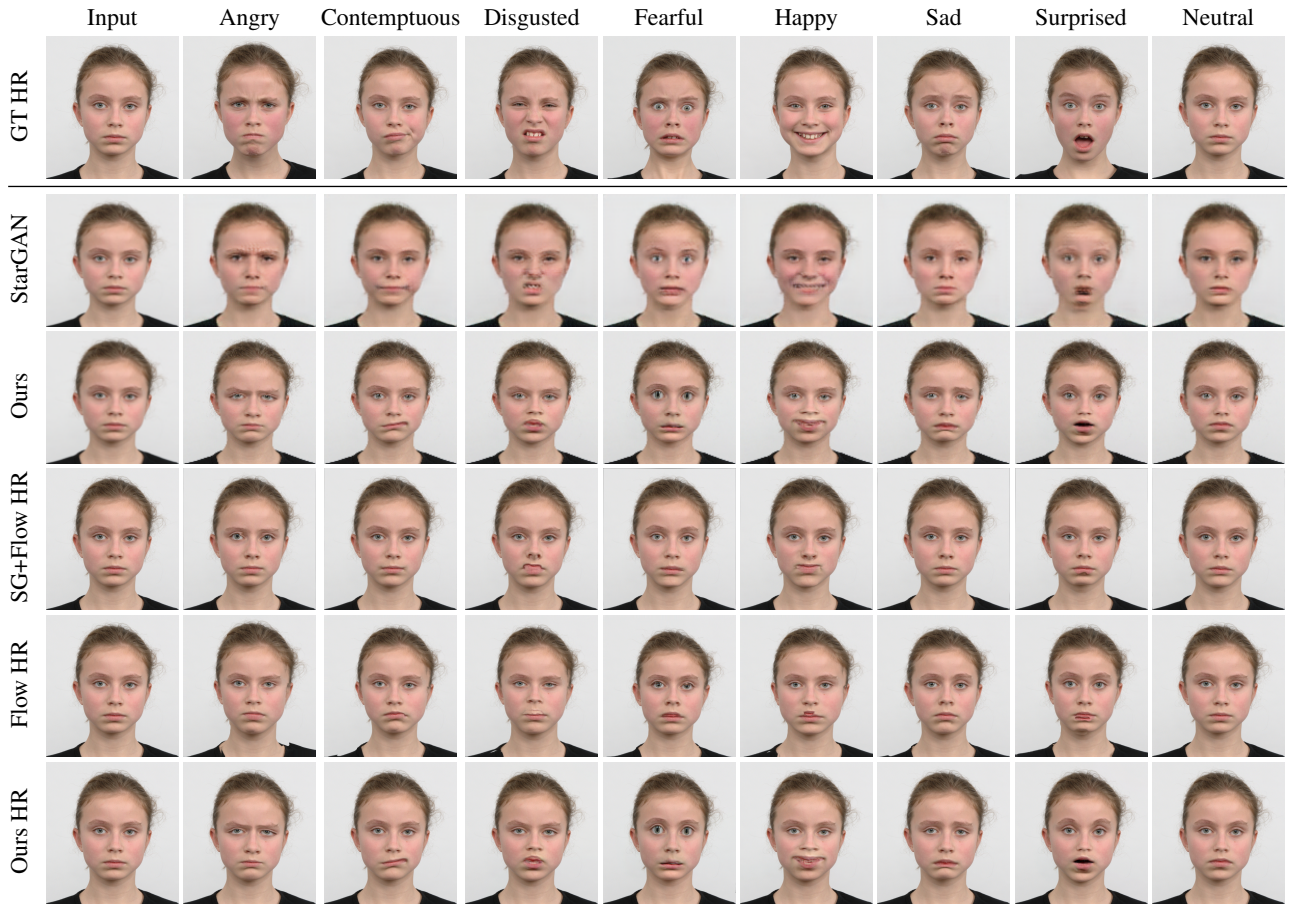

Figure 4: Comparison to previous work methods on the RafD dataset. From a given input image, first column, each method attempts to transfer the semantic attribute in its corresponding column. This dataset contains paired examples which we show in the top row. Our approach is able to edit the attributes of the input images while better keeping the face identity. (Zoom in for details)

Fig. 5. This is a cheap operation as it does not require to run the forward model for each new value of α, in contrast with previous methods that allow for partial edits [30]; this allows for edits to be performed at interactive speeds.

Please see Appendix A and E for additional qualitative results.

5.3. Quantitative results

Quantitative evaluation is challenging for our setting (especially with unpaired data). We provide two methodologies. The first is based on separately trained models and the second on a user study to estimate perceptual results. We train a classifier on the training data, to estimate quantitatively if the edited images have the required attributes. The classifier has the same architecture as the discriminator and is trained with the cross entropy loss of (eq. 4). It achieves an average accuracy of 82.46% on the real test data; we note that the classifier is indicative but should not be considered ground-truth, a fact confirmed by the user study. We also use a pretrained face re-identification model [31] to evaluate whether the edits preserve the identity. Results of both experiments are shown in Fig. 6. Even though our model is not able to transform the images as much as previous work, sometimes producing edited images which are not correctly classified, it is, in all cases, better at preserving identity.

We perform a user study on Amazon Mechanical Turk (MTurk) to evaluate the quality of the generated images, as shown in Tab. 2. We used 250 test images from the CelebA dataset and performed an edit based on a random attribute, using both our method and StarGAN. We conducted two experiments, one to evaluate the realism of the images, and another to evaluate whether the edited images contain the target semantic attribute.

In both user studies, the workers were randomly shown a single image at a time: an image by one of the methods or an unaltered original image. To evaluate the reliability of the workers, a number of easy to classify images were mixed with the data, and used as a control. We discarded images with fewer than 3 annotations. A simple majority voting scheme was used to determine the classification of each image.

In the real vs generated user study the worker had to an-

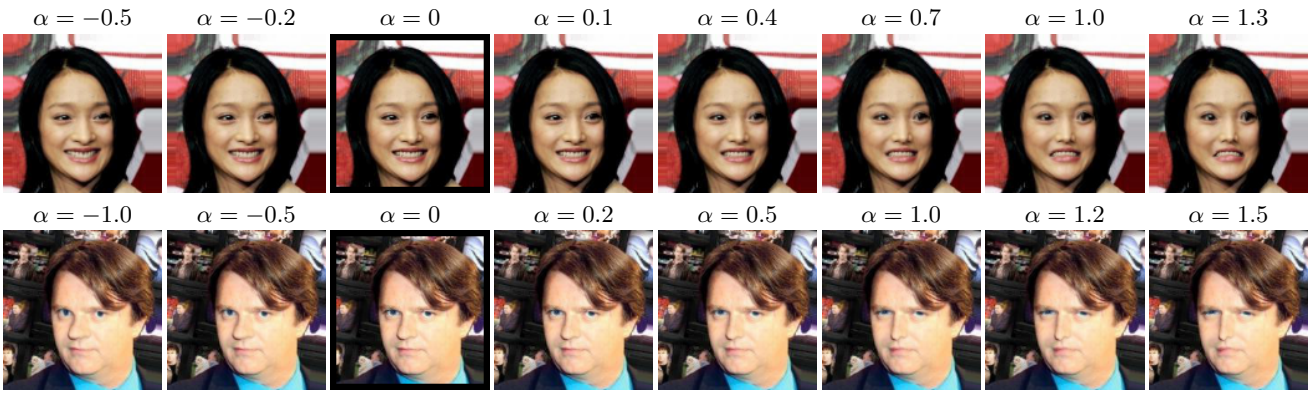

Figure 5: Partial editing with our model. A single warp is generated by our model, which is interpolated and extrapolated by scaling the magnitude of its values by α. The input image, $\alpha = 0$, is progressively edited in both directions.

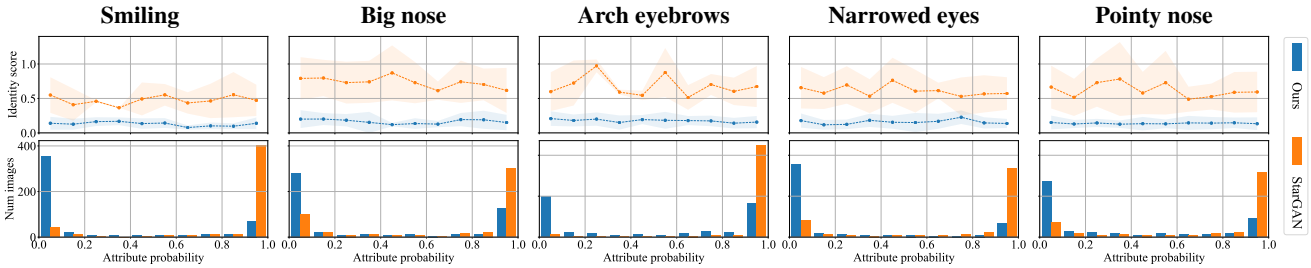

Figure 6: **Top:** Attribute classification accuracy (x-axis) vs re-identification accuracy (y-axis, lower is better, 1σ error bars). **Bottom:** Attribute classification accuracy (x-axis, larger is better) vs number of images (y-axis). Our model is unable to edit as many images as previous work, however it consistently produces edits that better preserve identity.

Model	Smiling	Big nose	Arch eyebrows	Narrowed eyes	Pointy nose	Mean	Real
StarGAN [6]	**87.50**	**84.00**	92.31	**87.50**	71.43	**84.25**	27.50
Ours	23.08	72.22	**96.15**	83.33	**76.19**	70.87	**86.21**
Real	95.45	47.82	69.57	78.79	35.0	66.94	100.00

Table 2: Human evaluation of the edits generated by the different models in [%], higher is better. The annotators consider the images generated by our model to be more realistic than previous work (far right column). Our method is also able to achieve the desired target edit, for most attributes, as classified by the annotators.

swer whether the image presented was real or fake. Typical failure cases for both models were shown to the worker before commencing the task, as examples of fake images. As shown in Tab. 2, the workers consider the images generated by our model to be more realistic than StarGAN.

For the semantic attributes, we asked the users whether the image contains the target attribute. To guide the workers, examples from training data were shown to highlight the differences between the attributes. In contrast to the results from the classification network, we achieve competitive results for most of the attributes, *i.e*. the workers correctly classified the images edited by our model in terms of the semantic attributes. We note that this part of the study might be less reliable since the users made errors on the real data suggesting some attributes were hard to judge.

6. Conclusions

This paper has introduced a novel way to describe semantic image edits from unpaired data using warp fields. We have demonstrated that, despite limitations on the set of edits that can be described using warping alone, there are several clear advantages to modeling edits in this way: they better preserve the identity of the subject, they allow for partial edits, and they are applicable to arbitrary resolutions.

There are several avenues for future work, including different parameterizations for the warps, *e.g*. in the form of velocity fields [4]. Additional intermediate representations that upsample well could be added to increase the model flexibility, such as local color transformations [10].

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[3] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[4] C. Ceritoglu, X. Tang, M. Chow, D. Hadjiabadi, D. Shah, T. Brown, M. Burhanullah, H. Trinh, J. Hsu, K. Ament, D. Crocetti, S. Mori, S. Mostofsky, S. Yantis, M. Miller, and J. Tilak Ratnanather. Computational analysis of lddmm for brain mapping. *Frontiers in Neuroscience*, (7), 2013.

[5] J. Chen, A. Adams, N. Wadhwa, and S. W. Hasinoff. Bilateral guided upsampling. *ACM Trans. Graph.*, 35(6):203:1–203:8, Nov. 2016.

[6] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] T. Dekel, C. Gan, D. Krishnan, C. Liu, and W. T. Freeman. Sparse, smart contours to represent and edit images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] E. L. Denton, S. Chintala, a. szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015.

[9] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky. Deepwarp: Photorealistic image resynthesis for gaze manipulation. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 311–326, Cham, 2016. Springer International Publishing.

[10] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118, 2017.

[11] A. Gilbert, J. Collomosse, H. Jin, and B. Price. Disentangling structure and aesthetics for style-aware image completion. In *2018 Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.

[14] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016.

[15] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. *arXiv preprint arXiv:1709.01118*, 2017.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[18] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[20] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and Emotion*, 24(8):1377–1388, 2010.

[21] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[22] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, December 2015.

[23] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276. ACM, 2001.

[24] L. Ma and Z. Deng. Real-time facial expression transformation for monocular rgb video. *Computer Graphics Forum*, 2018.

[25] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. I. Kim. Unsupervised attention-guided image to image translation. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

[26] U. Mohammed, S. J. Prince, and J. Kautz. Visio-lization: generating novel facial images. *ACM Transactions on Graphics (TOG)*, 28(3):57, 2009.

[27] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.

[29] T. Portenier, Q. Hu, A. Szabó, S. A. Bigdeli, P. Favaro, and M. Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Trans. Graph.*, 37(4):99:1–99:13, July 2018.

[30] A. Pumarola, A. Agudo, A. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[31] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[32] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[33] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner. Face2face: Real-time face capture and reenactment of rgb videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[34] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591. IEEE, 1991.

[35] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger. Deep Feature Interpolation for image content changes. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[37] R. A. Yeh, Z. Liu, D. B. Goldman, and A. Agarwala. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961*, 2016.

[38] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition*, pages 214–223, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[39] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 597–613, Cham, 2016. Springer International Publishing.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

Appendix

A. High-resolution results

Figure 7: Additional results of our model on high-resolution images. Our model predicts warps at low resolution that can then be resized and applied to high resolution images. The model is able to keep the content and identity at high resolution. However, due to the binary nature of the attributes, more than one attribute might be erroneously edited. Input images courtesy of flickr users Kenneth DM and Pedro Ribeiro Simoes.

B. Landmark locations

Figure 8: An example of the locations of the 49 face landmarks used in our model and described in section 4.2 of the paper. The location of the landmarks was chosen to provide useful control points for face warping. This landmarks are automatically generated by an internal neural network method.

C. Quantitative results

C.1. Attribute classification accuracy

Model	Smiling	Big nose	Arched eyebrows	Narrow eyes	Pointy nose	Mean
StarGAN	**93.12**	**74.03**	**94.43**	**79.10**	**83.17**	**84.74**
Ours	21.93	38.95	61.46	29.28	45.61	39.59
Real	91.51	80.47	81.16	86.58	72.56	82.46

Table 3: Quantitative comparison of the attribute classification accuracy on real and generated images on the CelebA dataset for the different models in [%], higher is better. As our method is restricted in the edits that it can do to the image, it falls behind to StarGAN.

C.2. Re-identification network description

For face re-identificaiton scores, presented in Fig. 6 in the paper, we use a Facenet model pretrained on the MS-Celeb-1M dataset [14]. This dataset consists of 10 million images and 100k unique identities. As both CelebA and MS-Celeb-1M where collected from publicly available internet images, we expect some overlap between both datasets. In all our experiments, we report the sum of the square difference of the network embeddings of the images, following the protocol recommend by the authors.

D. Interpreting warp fields: stretch maps

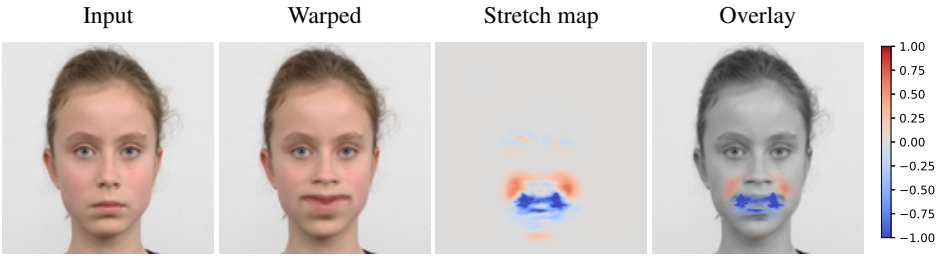

Figure 9: Stretch maps computed from the warp fields from the warp model, the mouth inpaint model is not used. We show the log determinant of the Jacobian of the warp, where blue indicates stretching and red corresponds to squashing. The values from the stretch maps can potentially be used to automatically determine which areas of the image have been stretched or compressed excessively by the network. Thus they provide an intuitive measure to detect unrealistic edits that could potentially be fixed with inpainting.

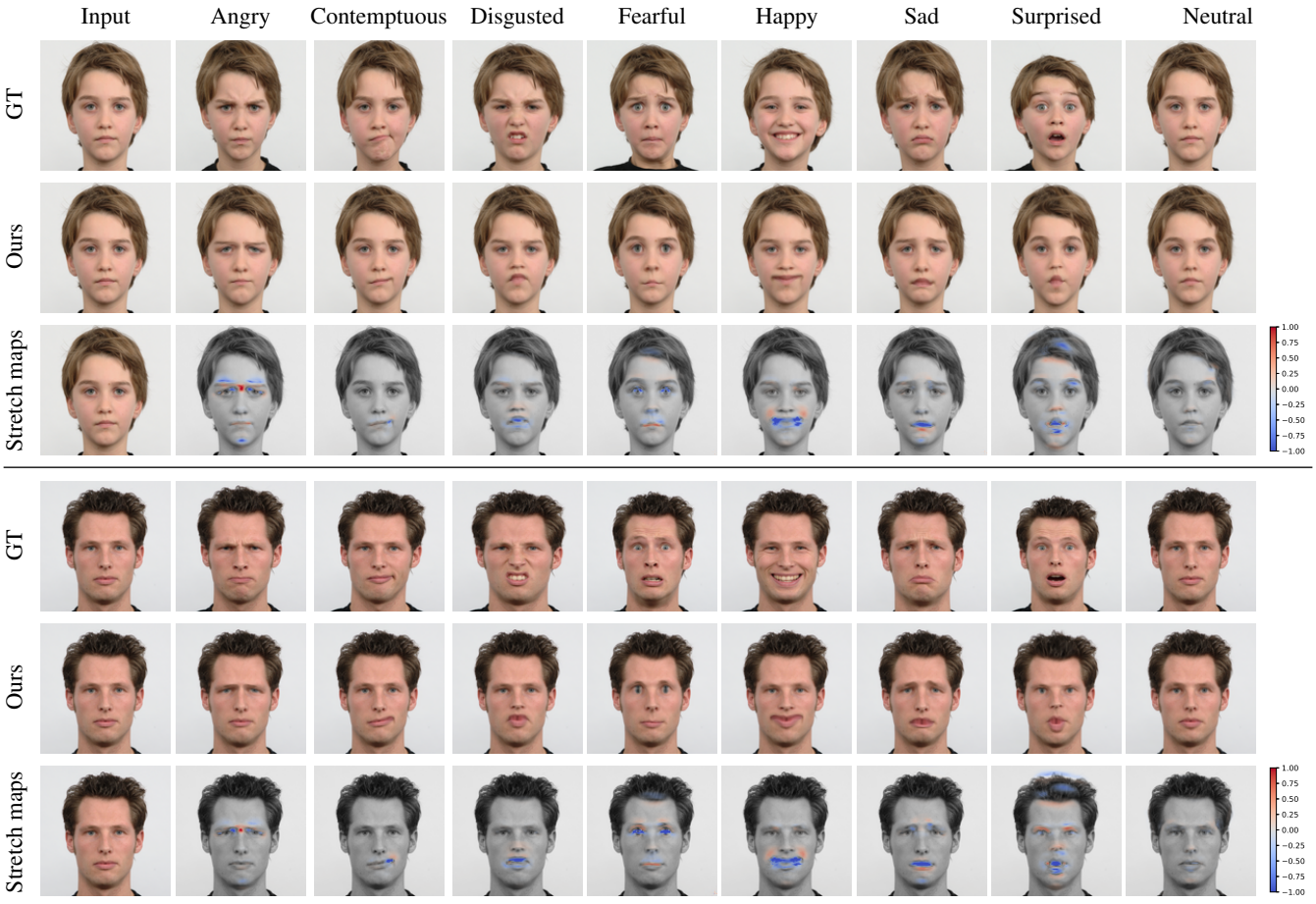

Figure 10: Results of our warp model, without mouth inpainting, on the RafD dataset. For each subject we show the ground truth (GT) on the first row, the result of our method on the second row and stretch maps computed from the warp fields on the third row (similar to Fig. 9).

E. Additional qualitative results on CelebA

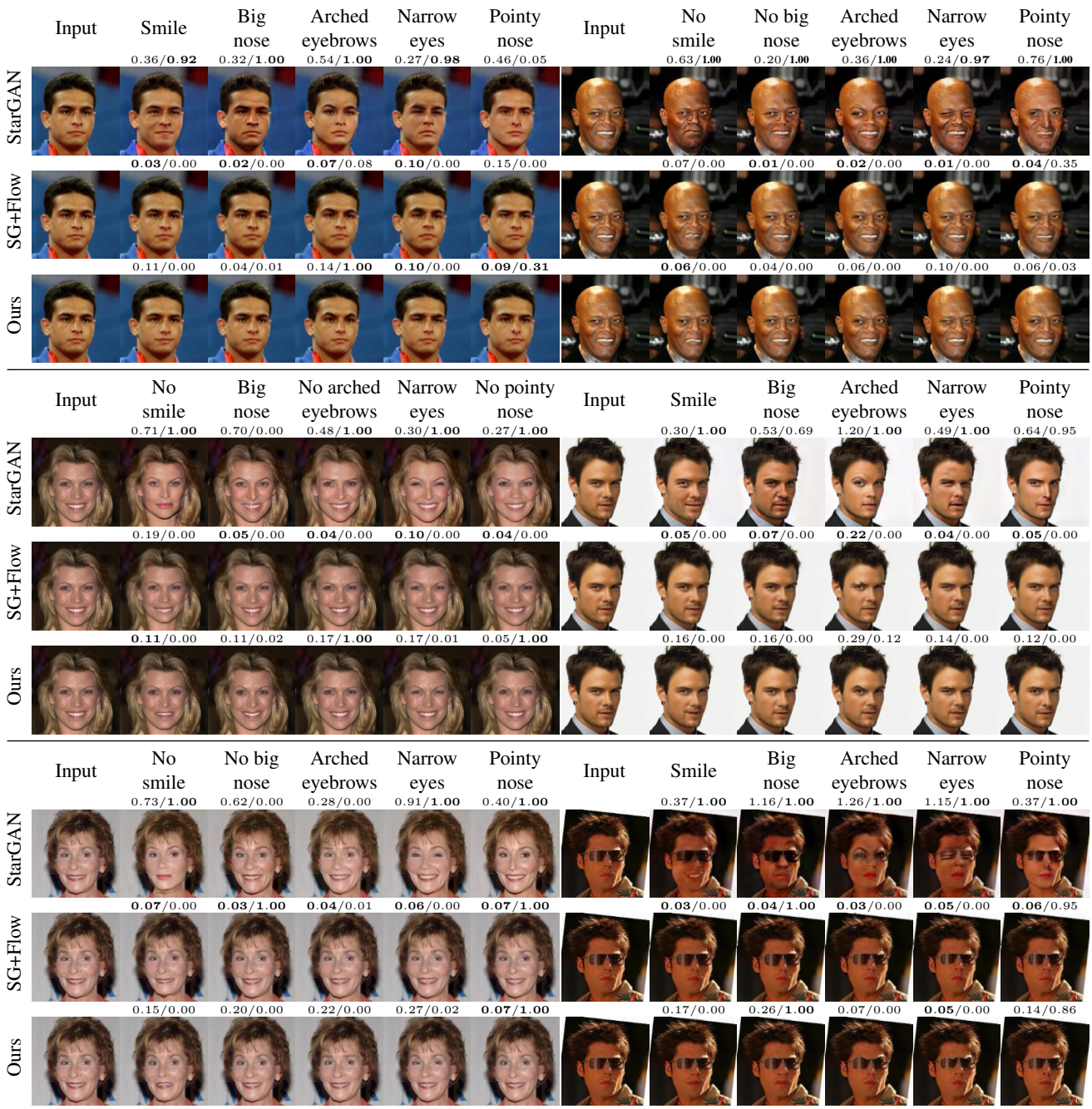

Figure 11: Comparison to previous work methods on the CelebA dataset. From a given input image, first column, each method attempts to transfer the semantic attribute in its corresponding column. On top of each image the re-identification score (lower is better) and the classification accuracy (higher is better) is shown as (id / cls).

F. Landmark based warping

We experimented with a landmark based method, as detailed in section 4.1 of the paper. The sparse displacements are defined for each of the 49 landmarks shown in Fig.8. A thin plates splines interpolation is used to produce a dense warp field from the sparse displacements. The networks for this model consist of 5 fully connected layers for the generator and 10 for the discriminator. The input to the generator consists of the landmarks x and y coordinates concatenated with the target labels. As the sparse warps are by construction smooth, we remove the smoothness losses. Large values of λ_{cls} produce unrealistic edits and visible artifacts, highlighting the lack of flexibility in the model.

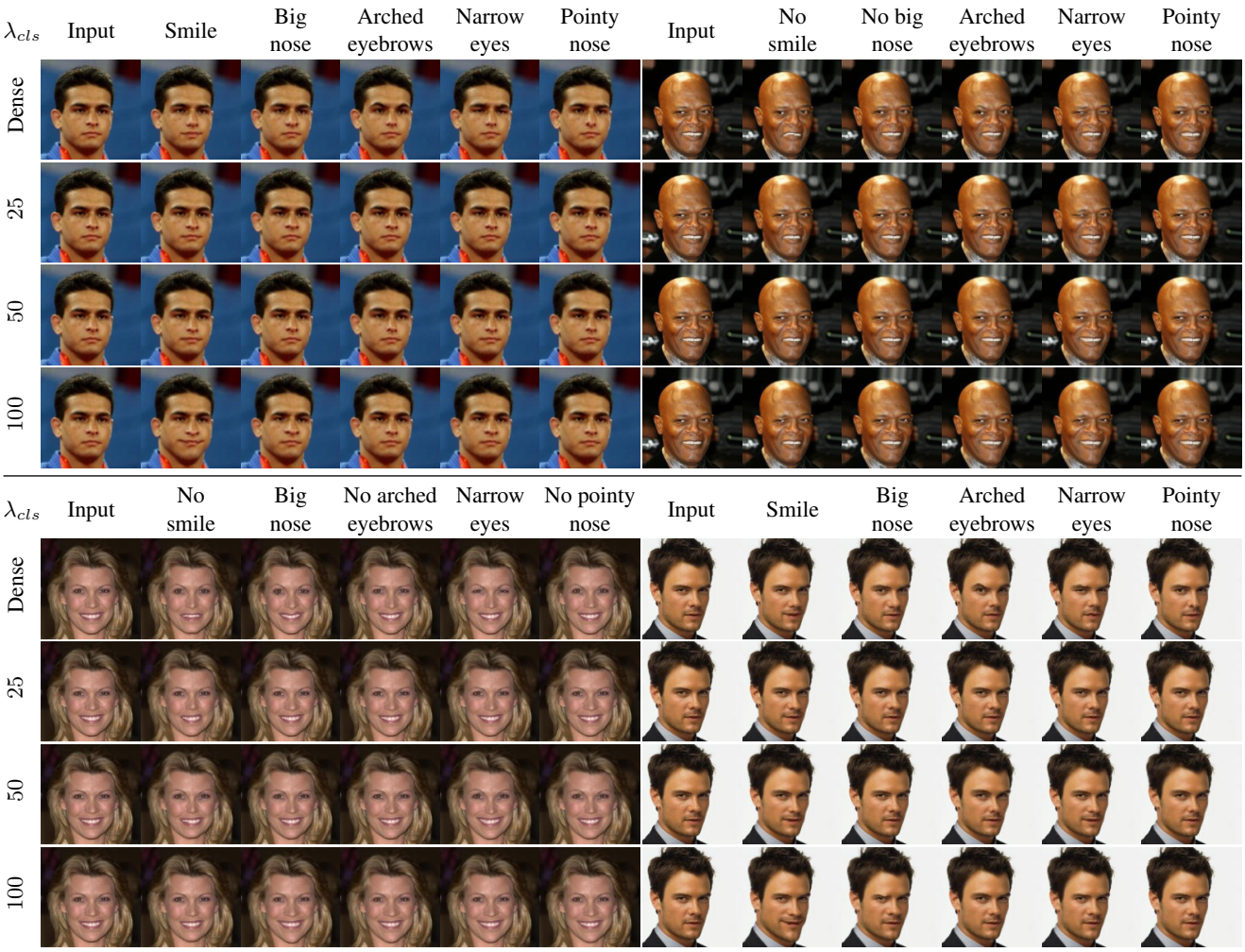

Figure 12: Results of the landmark based model on the CelebA dataset, for different values of λ_{cls}. Results from our dense model are also shown to more easily compare the results from the sparse model. The landmark based model is limited by the location of the landmarks, struggling with fine grained warps when not enough landmarks exist around an area, *e.g.* the eyebrows and the nose.

G. Parameter sweep for λ_{cls}

In this section we show both quantitative and qualitative results of changing the value for λ_{cls}, i.e. the weight for the classification loss. Increasing this weight leads to artifacts and loss of identity, without corresponding gains in terms of classification accuracy. This motivates our choice of $\lambda_{cls} = 25$.

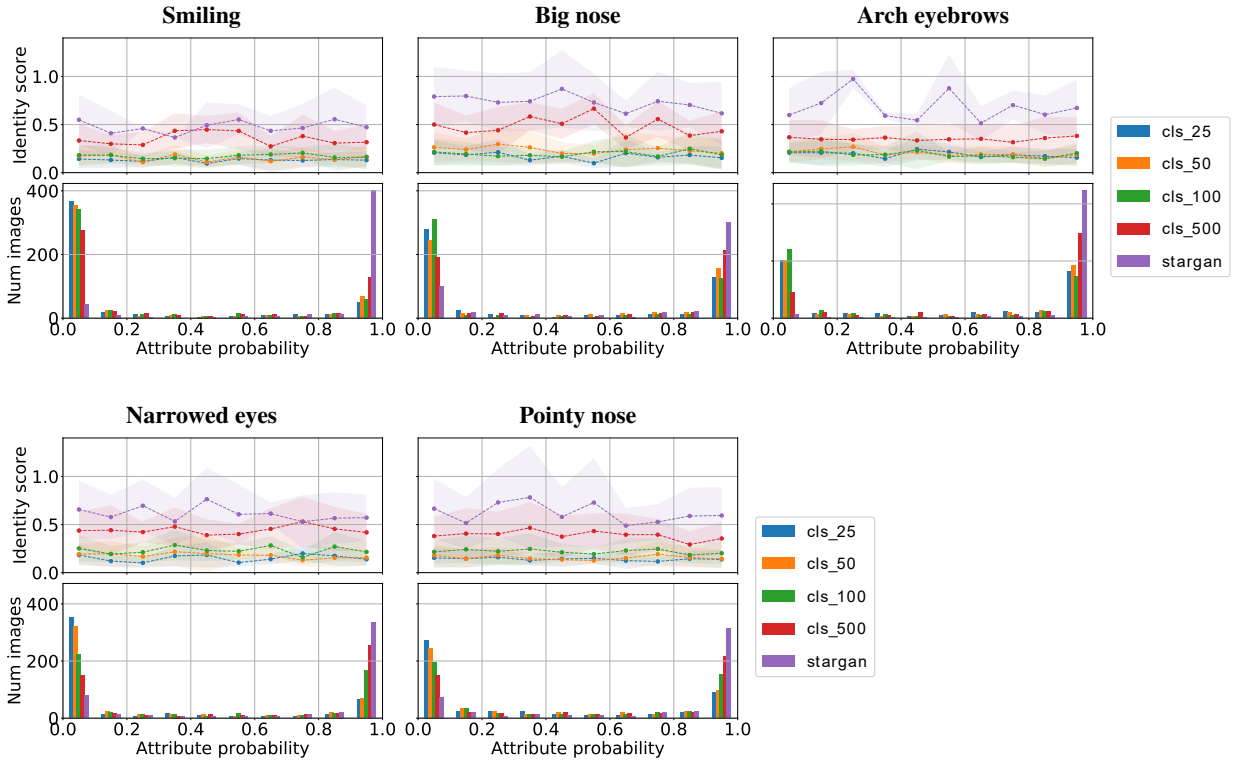

Figure 13: **Top:** Attribute classification accuracy (x-axis) vs re-identification accuracy (y-axis, lower is better, 1σ error bars). **Bottom:** Attribute classification accuracy (x-axis, larger is better) vs number of images (y-axis). Parameter sweep for the classification weight, λ_{cls}, where our model is still able to produce images that better keep identity for large values of λ_{cls}.

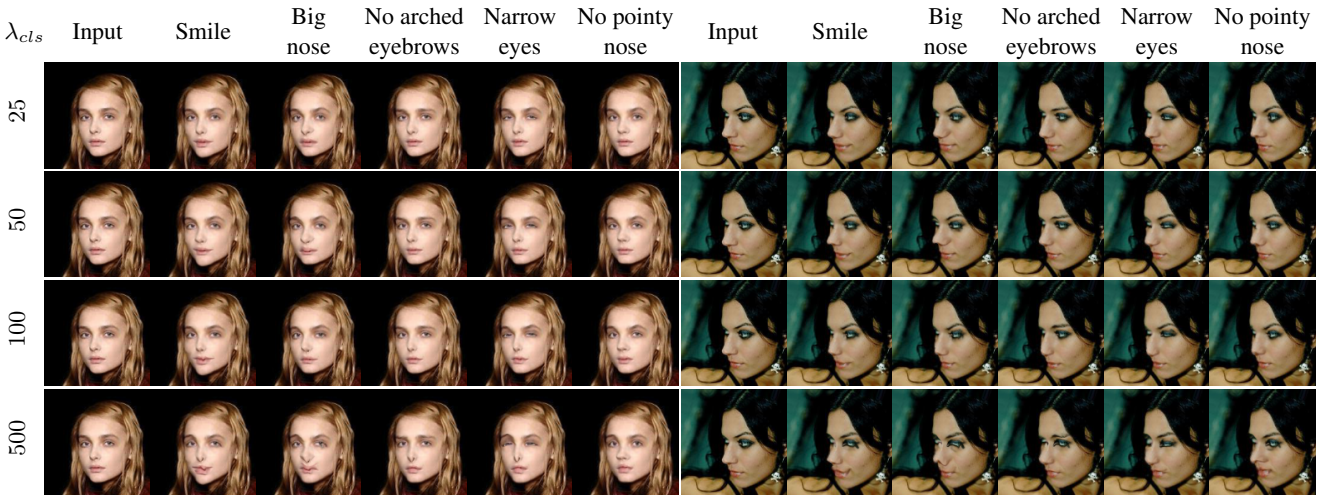

Figure 14: Results from our model on a parameter sweep for λ_{cls}. Large values of λ_{cls} produce unrealistic edits and visible artifacts, so we use $\lambda_{cls} = 25$ for all results in the paper.

H. Network architectures

Our architecture is based on the models in StarGAN, for the generator we replace all transpose convolution layers for bilinear resizing followed by convolution, and we replace all instance normalization layers with batch normalization. For the discriminator we use the architecture from StarGAN without any modifications. In both tables the following notation is used, N is number of output channels, K is kernel size, S is stride size, P is padding size and BN is batch normalization.

Part	Input $\rightarrow$ Output Shape	Layer information
Down-sampling	$(h, w, 3 + nc) \rightarrow (h, w, 64)$	CONV-(N64, K7x7, S1, P3), ReLU, BN
	$(h, w, 64) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S2, P1), ReLU, BN
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	CONV-(N256, K4x4, S2, P1), ReLU, BN
Bottleneck	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{4}, \frac{w}{4}, 256)$	Residual Block: CONV-(N256, K3x3, S1, P1), ReLU, BN
Up-sampling	$(\frac{h}{4}, \frac{w}{4}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 256)$	Bilinear resize
	$(\frac{h}{2}, \frac{w}{2}, 256) \rightarrow (\frac{h}{2}, \frac{w}{2}, 128)$	CONV-(N128, K4x4, S1, P1), ReLU, BN
	$(\frac{h}{2}, \frac{w}{2}, 128) \rightarrow (h, w, 128)$	Bilinear resize
	$(h, w, 64) \rightarrow (h, w, 64)$	CONV-(N64, K4x4, S1, P1), ReLU, BN
	$(h, w, 64) \rightarrow (h, w, 2)$	CONV-(N2, K7x7, S1, P1), Sigmoid

Table 4: Architecture for the generator network, G.

Part	Input $\rightarrow$ Output Shape	Layer information
Down-sampling	$(h, w, 3) \rightarrow (\frac{h}{2}, \frac{w}{2}, 64)$	CONV-(N64, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{2}, \frac{w}{2}, 64) \rightarrow (\frac{h}{4}, \frac{w}{4}, 128)$	CONV-(N128, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{4}, \frac{w}{4}, 128) \rightarrow (\frac{h}{8}, \frac{w}{8}, 256)$	CONV-(N256, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{8}, \frac{w}{8}, 256) \rightarrow (\frac{h}{16}, \frac{w}{16}, 512)$	CONV-(N512, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{16}, \frac{w}{16}, 512) \rightarrow (\frac{h}{32}, \frac{w}{32}, 1024)$	CONV-(N1024, K4x4, S2, P1), Leaky ReLU
	$(\frac{h}{32}, \frac{w}{32}, 1024) \rightarrow (\frac{h}{64}, \frac{w}{64}, 2048)$	CONV-(N2048, K4x4, S2, P1), Leaky ReLU
Output layer D	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (\frac{h}{64}, \frac{w}{64}, 1)$	CONV-(N1, K3x3, S1, P1)
Output layer C	$(\frac{h}{64}, \frac{w}{64}, 2048) \rightarrow (1, 1, n_d)$	CONV-(N(n_d), K$\frac{h}{64}$x$\frac{w}{64}$, S1, P0)

Table 5: Architecture for the discriminator and the classifier networks, D and C. All the down-sampling layers are shared by D and C.

I. Mouth inpainting

For the semantic edits that we are interested in, some might require inpainting, where a divergent warp would unocclude regions unseen in the input image (*e.g*. teeth). There are plenty of previous methods for inpainting [11, 27, 36] and this topic is not the focus of this work. Our solution consists of inpainting at a higher resolution a small part of the image.

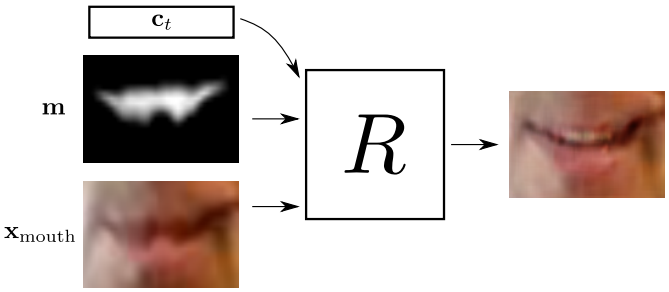

Figure 15: Overview of the generator network, R, for mouth inpainting. The inputs are a warped RGB image, $\mathbf{x}_{\text{mouth}}$, a soft mask with the mouth area, $\mathbf{m}$, and a target label, $\bar{\mathbf{c}}$. The generator synthesizes inner mouth details such as teeth.

We provide a simple solution for the particular case of the transition from mouth close to mouth open. If an edit requires such transformation, warping alone is unable to synthesize the mouth content, *e.g*. teeth or tongue. Thus, a network is trained to generate content for overstretched mouth areas as illustrated in Fig. 15.

A GAN generator is added, such that it takes as input the stretched mouth area, a soft mask of the mouth area and the target labels. In turn, the discriminator is fed real images of open mouths and labels. We use R to denote the generator network and S for the discriminator. Further details on the architecture for both networks in shown in Fig. 16.

An $L2$ loss on the pixels *outside* of the mouth area is added to preserve them

$$L_{mse} = ||(R(\mathbf{x}_{\text{mouth}}, \mathbf{c}_t) - \mathbf{x}_{\text{mouth}}) * (1 - \mathbf{m})||_2^2, \tag{13}$$

where $\mathbf{x}_{\text{mouth}}$ is the cropped mouth region and $\mathbf{m}$ is the cropped mask for the mouth region.

The losses that are used to trained the inpainting model are

$$\begin{aligned} L_S &= -L_{adv} + \lambda_{gp} L_{gp}, \\ L_R &= L_{adv} + \lambda_{mse} L_{mse}, \end{aligned} \tag{14}$$

where L_{adv} is an adversarial loss, L_{gp} is a gradient penalty, λ_{gp} and λ_{mse} are user defined hyper-parameters. For both datasets we use $L_{gp} = 10$ and $L_{\text{mse}} = 100$.

Inference If inpainting is needed, the mouth area in the warped image at the original resolution is cropped and resized. The new mouth is synthetized by the inpaint generator network and resized to the original mouth size. The mouth image is composited with the warped image using Poisson blending [28]. The inpaint discriminator, $S(\mathbf{x}, \bar{\mathbf{c}})$, is used to evaluate the realism of the mouth area before and after the inpainting operation. If the inpainted mouth is considered to be less realistic than the warped image, the inpainted region is discarded.

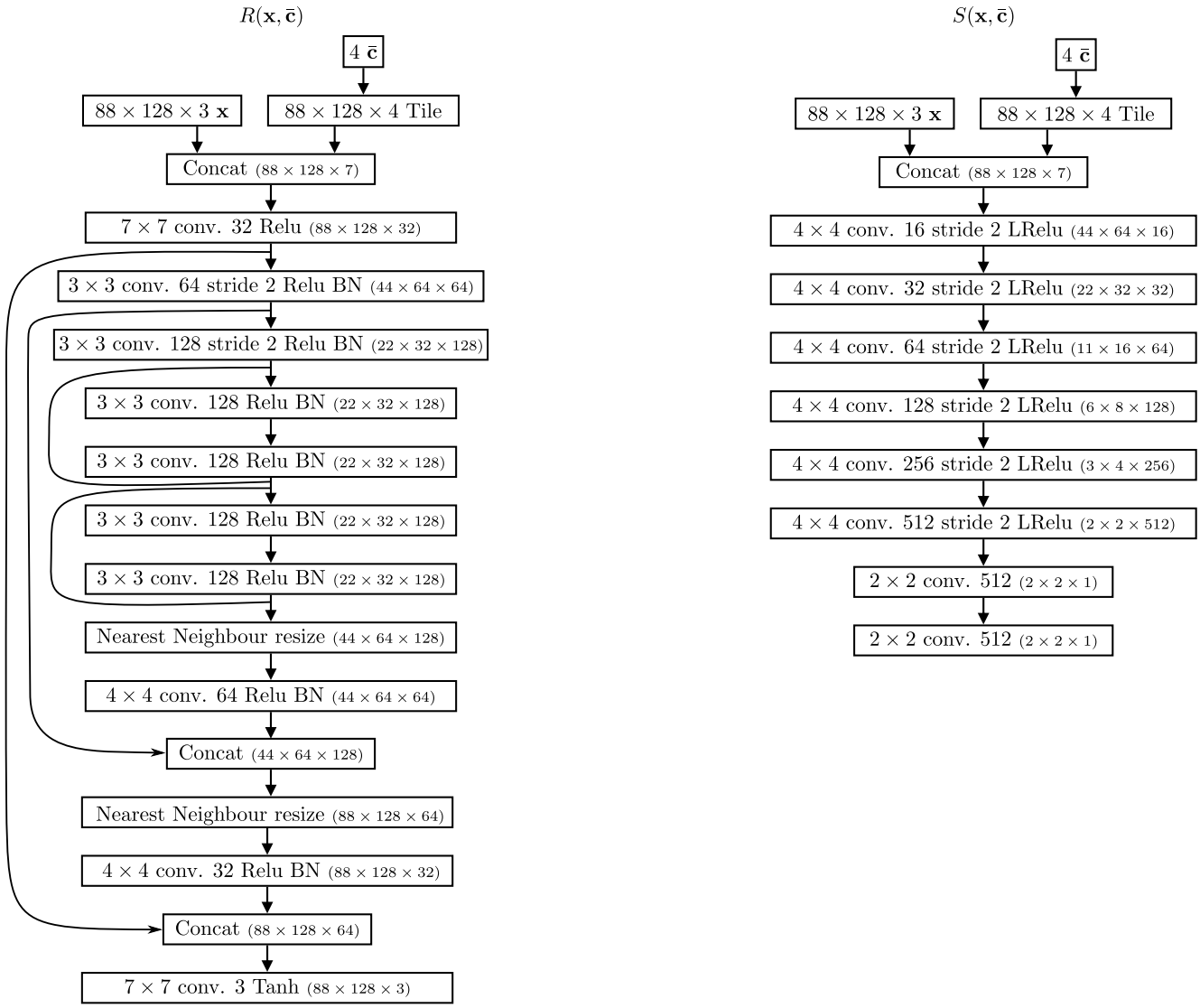

Figure 16: Architecture of the mouth inpaint generator network, $R(\mathbf{x}, \bar{\mathbf{c}})$, and the inpaint discriminator, $S(\mathbf{x}, \bar{\mathbf{c}})$. The output shape of each layer is denoted in parenthesis.

J. Ablation study

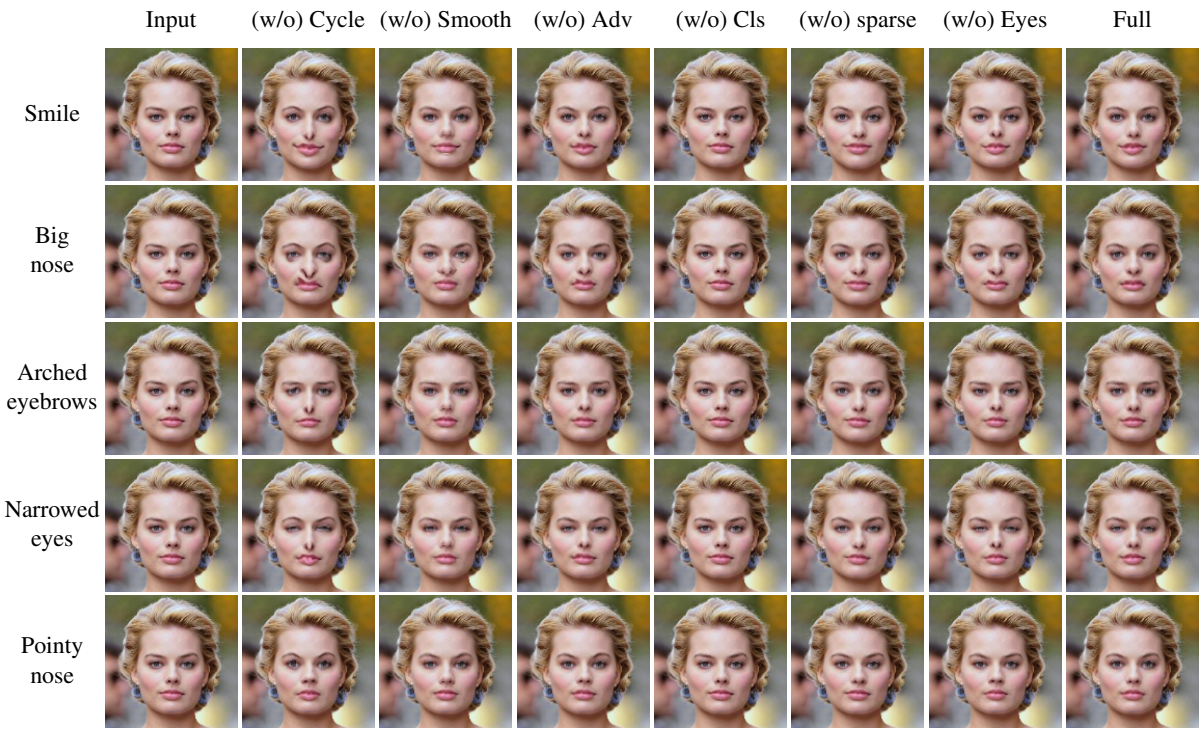

Figure 17: Ablation study, where we remove different losses in our model, the inpaint model is not used for these results. For each loss, (w/o) Cycle: significant artifacts are introduced, (w/o) Smooth: warps produce folding in the image, (w/o) Adv: unrealistic warps, (w/o) Cls: trivial solution on identity, (w/o) sparse: model changes parts of the image that are not needed, (w/o) Eyes: eyes are deformed in unrealistic ways.